UCRL- 100602
PREPRINT

# HASSLE-FREE AUDIT TRAILS:
# AUTOMATED AUDITS

Douglas R. Manatt

This paper was prepared for presentation at the
INGRESS User Association Meeting
New Orleans, LA

April 23-27, 1989

Lawrence
Livermore
National
Laboratory

# DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement recommendation, or favoring of the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

# Hassle-free Audit Trails:

# Automated Audits

Douglas R. Manatt
manatt@lll-winken.llnl.gov
Lawrence Livermore National Laboratory
P.O. Box 808, L-233
Livermore, CA 94550

## Abstract

*The origin and history of data in databases are often as important as the data itself. A full audit trail of database operations is the best record of a database's history. INGRES provides an audit facility to format journal file entries into audit records. This facility is cumbersome and difficult to use. I describe two INGRES Report Writer reports that take all the effort out of maintaining a complete audit trail.*

*To maintain an audit trail of changes to INGRES tables it is necessary to run AUDITDB individually on each table and store a record of the AUDITDB output. The INGRES manuals suggest how the audit records can be copied into INGRES tables for storage. Thus the maintenance of an audit trail consists of: creating tables to receive audit records, running AUDITDB, and storing the audit records into the tables. All this must be done for each table to be audited. My approach to this drudgery is to give it all to the INGRES system. Therefore, I present reports that generate command files to create the tables and run the audits. The only job left for a human is to submit the generated command files to the batch queue.*

**This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.*

## INTRODUCTION

When we first started using INGRES we had very few tables containing critical information. To audit these tables I wrote command procedures that ran AUDITDB every night and copied the audit results into tables. As our use of INGRES grew, we created many tables on different systems containing important data. This expansion of our INGRES usage and my desire to avoid plain and simple work lead me to revise the manner in which I handled audits, since I didn't want to spend all my time just updating audit scripts I automated the entire process. To this end, I wrote the following INGRES reports. They generate VMS DCL command files containing all necessary system and INGRES commands to generate and run an automated auditing system for all journaled tables in any INGRES database.

## OVERVIEW

The basic method is that for every database containing journaled tables, I create a second database to be loaded with the audit records. In this second data base each table retains its original name, but the attributes (columns) containing the audit information are prepreded to the tables actual columns. The audit is then run against each table indiyidually and the change records, if any, are copied to the second (audit) database. This process is repeated for each journaled database. This process decomposes to two major steps; the creation of the second database and its constituent tables, and the actual running of the audit and loading of the audit tables.

## NEWAUDIT REPORT

The system consists of two parts, first it is necessary to setup the table structures to receive the data from the audit records. This report "NEWAUDIT" reads the system tables of the database to be audited and outputs a command file to create and structure the secondary database. After running the command file generated by the "NEWAUDIT" report, all the tables to contain audit information have been created and wait empty for the first audit records to be copied in.

## INGAUDIT REPORT

The second operation, which is performed repeatedly, is to run the audits against the data tables and record the information. The report "INGAUDIT" again sorts through the systems tables looking for journaled tables and outputs a command file to run the audits and copy the audit trails into the waiting tables. When the procedure has finished it resubmits itself to run again the following day.

## RESTRICTIONS

This setup has several drawbacks. The audit information is not readily available along with the original table contents because the two different tables have the same name and are stored in different databases; this can be overcome using INGRES-STAR if necessary. In addition, since the audit process adds xxxx columns to the table data, this restricts the allowed column count of a journaled table to yyyy columns. Also, there are several operations which are very difficult to journal, table destruction, table creation and column addition or reordering are all operations which will completely confuse this automated process. Because of these restrictions, this system is most useful on systems without dynamic structure changes, i.e. mature applications. If you need a machine readable audit in a system with dynamic table structure changes, something besides INGRES will have to provide the solution.

## CONCLUSION

This technique is generally applicable, Report Writer can be used to generate any sort of program code desired, whether it be C, Fortran, DCL, OSL (4GL) or shell script.

I understand, in some programming shops, output is judged by the number of lines of code written per day. With tools such as these, you can afford to produce the daily special purpose procedures and codes that will appear to represent optimal job performance.